REMARKS

In regard to various references cited from searches done in other jurisdictions, the following statements will summarize these citations:

REFERENCES CITED IN AUSTRALIAN SEARCH:

X --- Derwent, IBM Research Disclosure, 10Jun1998.

X --- EP 80673, Sun Microsystems, 12Nov1997.

X --- W097/3303 -- Butts, Open Connection Systems, Incorporated, 9Oct1997.

Note: The Australian application was presented and issued as Australian Patent No. 770,160 on 27 May 2004. This case was filed on 23 October 1998 as Australian Application No. 10100/99.

REFERENCES CITED IN JAPAN:

1. Japanese Patent Laying Open No. 10-161976, which corresponds to its U.S. Application, 6,061,516, as issued.

2. Hiroki Kumagai "Practical CORBA & JAVA", JAVA WORLD, Vol.2 No.8 IDG Communications, August 1, 1998, pp. 130-135.

3. Current JAVA Programming --- Interview with Fujitsu Limited; JAVA Development Dept. DDJ
Dr. Dobbs Journal Japan Vol. 6, No. 12
Shoeisha Co., LTD., November 1, 1997, pp. 44-47.

In regard to Examiner's claim rejections under 35 USC Article 103, the Examiner has rejected claims 1-7, 10-20, 23-26, 28-34 and 37-40 for obviousness under 35 USC 103(a) as being unpatentable over Mills, et al. (W097/37303) in view of Matthews, et al., (U.S. 5,974,256).

Subsequently, when referring to Mills, it is assumed that the Examiner is actually referring to Butts, et al. (W097/37303). It is believed the Examiner has inadvertently used the wrong name, by referring to the citation by the agent's name, rather than by the Applicant's name. Thus, for clarity, Applicant will subsequently refer to the reference W097/37303 as "Butts".

> It should be noted that the Butts reference has a U.S. counterpart patent designated as U.S. Patent 5,754,830.

At this juncture, Applicant would traverse the Examiner's considerations that the above-rejected claims are obvious in view of Butts when combined with Matthews.

One of the major problems faced by the Applicant in the present invention was the problem of adapting a series of mainframe-based legacy applications, so that users could interact with the legacy applications via a modern distributed computer network.

> As the legacy applications are "customized" for each client, the use of a single emulator or emulation software would not provide a suitable interface for all the permutations of a legacy application.

Furthermore, an emulator does not allow the client to take advantage of the aesthetic and the operational benefits that stem from the provision of a Graphical User Interface.

However, the need to build multiple Graphical User Interface applications, one for each individual client, would have been a time-consuming and very expensive exercise for the Applicant.

As will be noted, the invention claimed in the independent claims provides an application which receives original legacy source code as an input, and then generates further source code which is compiled to create an executable application capable of operating on a distributed computer network to interface with the legacy application.

Therefore, Applicant's claimed invention resides --- in the provision of an application that is capable of utilizing legacy source code and converting the legacy source code into an executable application that may be run on a distributed network computer.

Applicant's invention provides a method and computer program and system for the automatic creation of a software program for interfacing with existing legacy applications. In particular, the invention seeks to extend the functionality of existing legacy applications (that may have traditionally run in an environment comprising a centralized computing resource interconnected to a series of computer terminal devices) to a

network environment (comprising a system of distributed, interconnected network computing resources).

Importantly, the present invention enables access to the full functionality of the legacy software application by enabling interconnection between the computing resource and the executable code of the legacy software application. Access to the legacy application may therefore be just as complete from the network environment as from the usual computer terminal device connection.

This is achieved by utilizing translatable source code for the initial legacy software application to produce a series of software components that are executable by the computing resources in the network environment.

In the preferred embodiment, the software application includes a plurality of interface specification definitions which include a definition of a screen format for a user interface, and a series of interactive screen software components which are generated from the interface specification definitions describing the screen formats.

Using these components, the computing resource is able to connect with the executable code of the legacy software application.

Figure 3 best illustrates the operation of the present invention. In this particular example, ISPEC components (being interface specification definitions in the LINC 4 GL language)

are generated from the translatable source code of the legacy ISPEC definition (see Fig. 4 which discloses the generation of an ISPEC component "Make Sale"). A plurality of ISPEC components are generated in this way, that can reflect the full functionality of all the ISPEC definitions provided in the translatable source code of the legacy application.

Further, ISPEC views corresponding to various screen views are created utilizing the components. The ISPEC view components 42-44 display their form and save any user input in a corresponding ISPEC object 46-48 on the generated client. The ISPEC objects are then passed on to the LINC environment's transaction manager 35 for sending onto the remote LINC application 23 for processing.

In summary, the present invention provides a method for readily extending the utility of a 4GL (preferably) application into a Web environment. The extension is achieved by taking the originally-designed 4GL application and providing a translator to create a series of codes that can be used to provide the functionality for interaction with the previously generated 4GL application over the Internet interaction (or at least interaction from a browser) therefore can be achieved with a previously-constructed host application in a substantially simplified and automated manner.

It may be noted that while claims 1, 14 and 28 claim the invention in slightly different terms to claim 27, they are nevertheless related to cover the same invention. The "template" definitions referred to in claim 27 are equivalent to the interface definitions

referred to in the preamble of the specification and in claim 2.

As previously indicated, the cited references do not disclose the method and the system of the present invention. In particular, they do not disclose the idea of taking translatable source code (or template definitions) and providing a translation of the source code or template definitions into components which can be utilized in a network resource for interacting with the executable code of the original legacy application. Thus, this enables a full functionality of the legacy application to be accessed using a browser and/or over the Internet.

The Butts reference (W097/37303) does appear to disclose a legacy application interface which allows for Web-based applications to interface with the legacy system. There is some discussion in the preamble of the Butts application of traditional ways of connecting legacy systems to network terminals. For example, one conventional method is to execute a terminal emulator program on a client that connects directly to a host legacy system using a TCP/IP socket connection. Another conventional method is to provide connection through a Web browser application by translating standard legacy data flows into 8GML pages.

The Examiner should note that neither of these methods include the idea of creating functional objects from the translatable source code of the original legacy application and then using those objects to interact with the executable code of the original legacy application.

It may further be noted that the invention disclosed in the Butts reference merely discloses a slight variation on the traditional methods of connecting with the legacy applications.

It does not disclose the idea of translating templates of the legacy application into components as is done in the present invention.]

> The Butts reference (W097/27537) merely appears to disclose a processor which is able to execute instruction sets received from a network or a local memory. This is a processor that can execute two different instruction sets --- but this does not disclose nor contemplate the method disclosed in Applicant's present invention.

The Butts reference discloses a legacy application interface which allows the user to interface with a legacy system via the Internet (or other suitable network). The Butts invention achieves this aim by providing a JAVA applet which is arranged to execute or operate within a web browser. Then, the JAVA applet emulates a terminal screen which would traditionally appear on a "character terminal".

> By definition, an emulator presents a screen to the user that is substantially similar, if not identical, to the terminal screen used on a character terminal. Therefore, the emulator is inherently limited to the functionality and aesthetic and operational features of the original character terminal. Moreover, emulators must pre-suppose certain patterns of behavior and certain commands. That is to say, an emulator is not necessarily suitable for all the permutations of a particular legacy application. It may be noted that legacy applications which have been

"customized" for a particular client may also require a "customized" emulator.

In summary then, there is no teaching in the Butts reference of a system which takes source code and utilizes said source code to produce a series of executable software components for providing the functionality for interaction with a legacy application.

Now, as to the Matthews reference, U.S. 5,974,256 --- this discloses a method for translating a Graphical User Interface resource data file into JAVA code. The invention disclosed in Matthews only teaches a translation tool which migrates or maps Graphical User Interface resource information from a Windows™ based resource file to JAVA native code --- to facilitate the migration of existing native applications.

In other words, the invention described in Matthews only translates sections of code which are explicitly related to the Graphical User Interface resource file of a particular application from one specific environment (i.e., Microsoft Windows™) to an operating system independent language --- namely, JAVA.

The Graphical User Interface resource files in a Windows application only represent a small proportion of the total code required to create and interface a Windows Graphical User Interface application. This is readily admitted in Matthews, at column 9, line 63 to column 10, line 4, where it is stated, in part "a set of 'work code' must still be

incorporated in order to provide a working interface".

Matthews does not suggest or teach towards the desirability of creating an executable application from legacy source code. Matthews is wholly concerned with a system that maps Windows-based Graphical User Interface resource files JAVA code.

> There is no teaching, implicitly or explicitly, that such a tool may be used to create an executable application that can interface with the legacy applications. Matthews teaches an invention that is merely the first step in migrating a Graphical User Interface from a Windows-based environment to a JAVA environment.

It now appears to be quite evident that there is no enabling disclosure in either of the prior art reference which would cause a skilled engineer to be led towards Applicant's claimed invention. Further, there is no teaching in either reference which would lead a skilled engineer to combine the references.

> As is indicated in the hereinbelow cited case citations, it should be indicated that it is improper for the Examiner to combine a second reference with a first reference if there is no indication in the first reference that such technology from the second reference would be desirable.

For example, in the case of *In re Jones*, 958 Fed.2d, p.347, 21 USPQ2d, pp.1941,1943 (Fed.Cir 1992), it was stated as follows:

> Before the PTO may combine the disclosures of two or more prior art references in order to establish *prima facie* obviousness, there must be some suggestion for doing so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art.

Further, in the case of *Uniroyal, Inc. v. Rudkin-Wiley Corp.*, 837 F.2d p.1044, and 5 USPQ2d, p.1434 (Fed.Cir 1988), it was stated: --

> When prior art references require a selected combination to render obvious a subsequent invention, there must be some reason for the combination other than the hindsight gleaned from the invention itself. Something in the prior art as a whole must suggest desirability, and thus, the obviousness of making the combination. It is impermissible to use the claims as a frame and the prior art references as a mosaic to piece together a facsimile of the claimed invention.

Moreover, even if such references were combined, it will be noted that the combined references do not teach the step of: --- "utilizing said source code to produce a series of software components for providing the functionality for interaction with the executable code, said components being executable by at least one of said computing resources . . . .".

> Therefore, it is now emphasized that Applicant's claim 1 cannot be obvious in light of the references to Butts and Matthews. Thus, Examiner's conclusions at

paragraph 5 of his Office Action should no longer be considered allowable objections.

In regard to Examiner's paragraph 6 and 7 of his Office Action with respect to claims 8, 21, 35 and with respect to claims 9, 22 and 36, which have been rejected under 35 USC 103(a) for obviousness over Butts (Mills) in view of Matthews, in view of Apte (U.S. 6,662,236 B1), plus mention of Harold regarding using component methods in an applet.

As noted by Examiner, Butts (Mills) does not teach a series of software components executable by scripting language running on the network computing source ---- where Examiner says that Apte teaches a series of software components as executable by scripting language running on the network computing resource, and that it would be obvious to apply the teaching of Apte to the system of Butts (Mills).

There is no warrant for trying to insert teachings of Apte or that of Harold into the Butts (Mills) reference, since there is nothing in Butts to indicate that such technology should be inserted therein. Further, this appears to be a situation where the Examiner is re-inventing Applicant's system after his viewing of various of the cited references.

Further, it should be noted that these claims which are dependent on the main independent claims which Applicant has emphasized cannot be considered to be obvious from the cited references.

In regard to Examiner's paragraph 8 of his Office Action, where claim 27 is rejected under 35 USC 103(a) for obviousness over Matthews, in view of admitted prior art (APA), wherein Examiner indicates that the Matthews reference does not teach a 4GL legacy application, but that there is earlier prior art which teaches 4GL as one type of legacy application, in which the system of Matthews could migrate such a 4GL legacy application. Here it should be noted that claim 27 involves the utilization of template definitions to produce a series of software components, said components being executable by at least a computing resource in a network environment comprising a system of distributed, interconnected network computing resources -- and no such specific teaching could certainly be found in the cited references.

In view of the situation that the cited references to Butts and Matthews cannot be considered having the teachings which would cover Applicant's independent claims, and further in regard to the impropriety of combining such said references to try to re-invent Applicant's method and system from enumerable cited references, it should be understood that Applicant's claims should be taken as a whole in their entirety, where as such, they indicate a functionality which cannot be provided for by the cited references, neither in the independent claims, nor in the dependent claims involved herein.

As a result, it is respectfully requested that Examiner consider Applicant's claims as a whole in their entirety and subsequently provide a timely Notice of Allowance therefor.

Respectfully submitted,

By Alfred W. Kozak

Alfred W. Kozak
Reg. No. 24,265
(858) 451-4615
(949) 380-5822

Certificate of Mailing (37 CFR 1.8a)

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: MAIL STOP AMENDMENT, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Patti S. Freddy

Date: June 14, 2004 Patti S Freddy